爱创课堂前端培训

CSS

第二天

班级： 爱创课堂 7 期

日期： 2017 年 4 月 14 日

目录

一、复习

1、html 杂项

- 注释：ctrl+/
- 转义字符：

```
通用格式：&关键词;
空格： 
大于号：&gt;
小于号:&lt;
版权信息:&copy;
花符号：&amp;
```

- 废弃标签：font,b,i,u,em;

2、css 基础知识

- css（cascading style sheet）层叠式样式表。css2.1 版本。
- css 作用：1.设置盒子样式用于布局 2.文字样式
- css 书写位置：

  1.内嵌式（嵌入式），书写在 head 标签里的一个 style 标签里。

  2.行内式，书写在标签里的 style 属性里。

  3.外链式
- css 一些属性：

  1.盒子属性：

```
width:宽
height:高
background-color:背景颜色
border:1px solid #eee;
```

  2.文字属性

```
font-family:字体
font-size:字号
color:颜色
```

- css 一些杂项

  键值对：k:v;

  分号的重要。

2、css 选择器

- 标签选择器：能够选中所有同种标签。作用：重置、清空默认样式。

```
p{
   color:#f00;
}
```

- id 选择器：id 是唯一的。命名规则，严格区分大小写，是字母开头，可以数字，下划线，横线。

```
#box{
    font-size:20px;
}
```

- 类选择器：能够选中具有相同类名的一些元素。

```
.box{}
```

- 通配符：*
- 后代选择器：

```
.box p{} //表示.box 后代中的 p
```

- 交集选择器：

```
.box.para{} //表示既有.box 类又有.para 类
```

- 并集选择器：或的意思。用逗号隔开选择器。

```
p,div,h3{}
```

二、css 继承性和层叠性

1、css 继承性

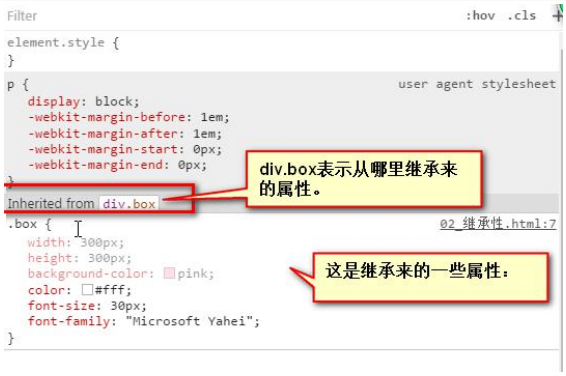
继承性：css 中，给祖先设置一些样式，后代元素中会继承这些。

```
    <div class="box">
        <p>我是一个段落</p>
        <h3>我是三级标题</h3>
        <div>
            <p>我是里边的 p</p>
        </div>
    </div>
```

给 div.box 设置样式，看看后代可以继承哪些属性

```

```



继承：**能够继承文字属性，不能继承盒子属性**。

文字属性：font-系列，color , line-系列，text-系列

盒子属性：width,height,background-系列，border-系列，浮动，定位

继承性是比较好的一个属性。可以简化咱们 css 代码。

```
        body{
            color:#333;
            font-size:14px;
            font-family:"Arial","Microsoft Yahei","SimSun";
        }
```

层叠性体现 2：继承性。

2、css 层叠性

不同的选择器都选中同一个元素，设置相同的属性，那么我该听谁的？

这个问题涉及权重的概念：

权重：选择器的针对性不同，针对性越强权重就越高。

权重的比较：id 选择器 > 类选择器 >标签选择器。

层叠性：不同的选择器同时选中了一个元素，权重高的会层叠掉权重低的属性。

问题：看看 p 的文字显示什么颜色。

```
        p{
            color:orange;
        }
        .para{
            color:red;
```

```
        }
        #para{
            color:green;
        }
```

最终 p 显示为绿色：

①如果选中了这个元素。计算选择器的权重。

（1 个 id = 255 类。）

计算方法：（id 选择器的数量，类选择器的数量，标签选择器的数量）

先比较 id 选择器的数量，id 选择器数量多的权重高；

id 数量相同，就比较类选择器的属性，类选择器的数量多的权重高；

id 和类选择器数量相同，就比较标签的数量，标签选择器数量多的权重高；

id 和类，标签选择器的数量都相同，就看谁写在最后，就是显示谁的样式。

```
    <div class="box1" id="box1">
        <div class="box2" id="box2">
            <div class="box3" id="box3">
                我是 div，看看我显示什么颜色？
            </div>
        </div>
    </div>
```

我们的 css 样式：

通过比较，第二个样式，id 数量为 1，权重最高。

最终显示颜色为：

有如下样式：最终显示什么颜色？（因为他们都是(1,2,0)，所以最终显示样式看谁写在最后就显示谁的样式。）

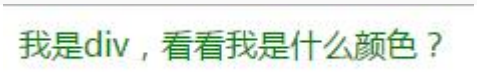
```
        #box1 .box2 .box3{
            color:orange;
        }
        .box1 .box2 #box3{
            color:yellow;
        }
        .box1 #box2 .box3{
            color:green;
        }
```

②没有选中元素，样式看继承性。

没有选中元素有权重，只是权重为 0.没有办法比较。看继承性

样式如下：都没有选中我们元素，继承样式看 html 结构中的距离远近，遵循**就近原则**，最终继承样式是距离最近的属性。

```
        body{
            color:yellow;
        }
        .box1{
            color:red;
        }
        .box1 .box2{
            color:green;
        }
        #box1{
            color:orange;
        }
```

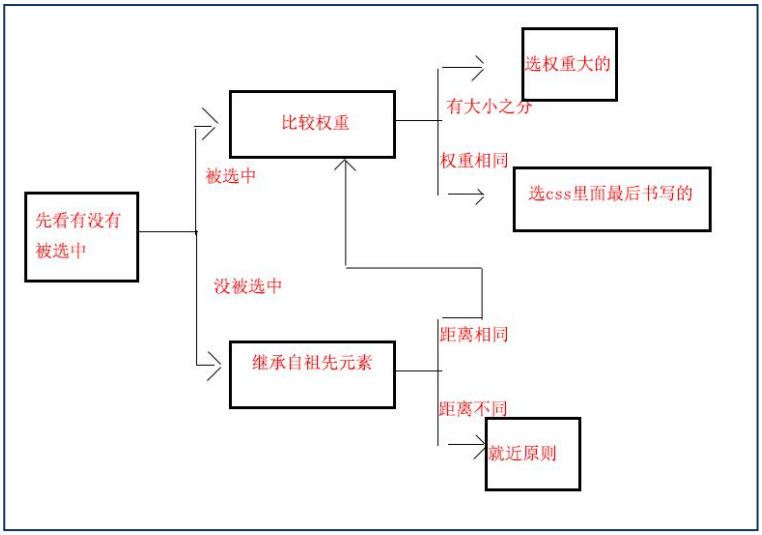


如果距离相同，就比较 id 选择器的数量，id 选择器数量多个权重高；

如果 id 数量相同，比较类选择器的数量，类选择器数量多的权重高；

如果 id 数量，类选择器的数量相同，就比较标签的数量，标签数量多的权重高；

如果他们数量都相同，就看 css 样式书写的顺序，谁在最后就显示谁的样式。



③!important

!important 加载属性值的后面，分号的前面。

选中元素时，!important 权重最高。

没选中元素，不影响就近原则。

!important 它只能调高一个属性的权重，不是所有选择器的属性。

```
        .box1 .box2 #box3{
            color:yellow;
        }
        #box1 #box2 #box3{
            color:red;
            font-size:40px;
        }
        div div div{
            color:green !important;
            font-size:20px;
        }
```

我是div，看看我是什么颜色？

三、文字属性

css 属性：文字属性，盒模型，背景，浮动，定位

1、color

color：文字颜色。

属性值：单词表示法，十六进制表示法，rgb 表示法。

实际工作中：设计会给我们提供。

现阶段需要自己手动吸取文字颜色。

2、font-size

font-size：字号。

单位：像素 px。

现阶段我们需要自己获取字号。自己写一个文字进行匹配。

3、font-family

font-family:字体

中文字体：微软雅黑，宋体

英文字体：黑体

```
font-family:"Arial","Microsoft Yahei","SimSun";
```

4、font-weight

font-weight:文字加粗。

属性值：数字表示 100-900 或者是单词表示。

normal：数值是 400；

bold：数值是 700。

5、font-style

font-style：文字是否倾斜。

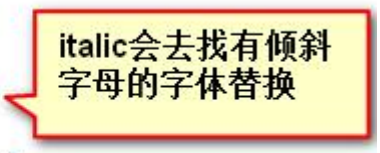
属性值：normal，italic(英文会自动找有倾斜字体的字母替换)常用。oblique（简单的倾斜）

```
        .italic{
            font-style:italic;
        }
        .oblique{
            font-style:oblique;
        }
```

this is a paragragh,normal

this is a paragragh,italic

this is a paragragh,oblique



6、line-height

line-height:行高。

行高：一行文字实际占有的高度，在行高内文字垂直居中。

表示方法：像素表示法和百分数表示法。

行高尽量写在字号的后面，因为用百分数表示时，是和字号比较的。

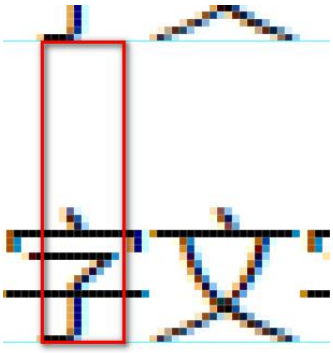
```
            font-size:20px;
            line-height:200%;
```

单行文本的垂直居中：行高等于盒子的高度。

```
        .para{
            width:500px;
            height:100px;
            border:1px solid #ddd;
            font-size:20px;
            line-height:100px;
        }
```

这是我们文字

行高的测量：①从上一行文字的底部到下一行文字的底部。



②匹配方法（写两行文字进行）

font 复合属性写法：

```
font:italic bold 20px/200% "Microsoft Yahei";
```

文本控制属性：text-align,text-indent,text-decoration

7、text-align

text-align:文本左右居中。

属性值：left,right,center。

默认属性值：left（文本居左。）

```
        .left{
            text-align:left;
        }
        .center{
            text-align:center;
        }
        .right{
            text-align:right;
        }
```

文本水平居中：text-align:center;(和文本是否是单行没有关系)

8、text-indent

text-indent:文本缩进。

文本缩进 2 个汉字：

```
text-indent:2em;
```

像素表示法：

```
text-indent:60px;
```

百分比表示法：表示是父盒子的宽度百分比。

```
        .box{
            width: 700px;
            height:500px;
            border:1px solid #ddd;
            font-size:20px;
        }
```

```
        .box p{
            text-indent:10%;
        }
```

9、text-decoration

text-decoration:文本修饰。

标签默认值是没有下划线的（a 标签除外）。

属性值：none；underline；

去掉 a 标签的下划线：

```
        a{
            text-decoration:none;
        }
```

四、盒模型

1、盒模型初步

一个真正的盒子应该有：宽度，高度，内边距，外边距，边框。

可以写内容的区域：

width：内容宽；

height:内容高

盒子占有的区域：

盒子实际占有的宽 = 内容宽 + 内边距*2 + 边框*2

盒子实际占有的高 = 内容高 + 内边距*2 + 边框*2

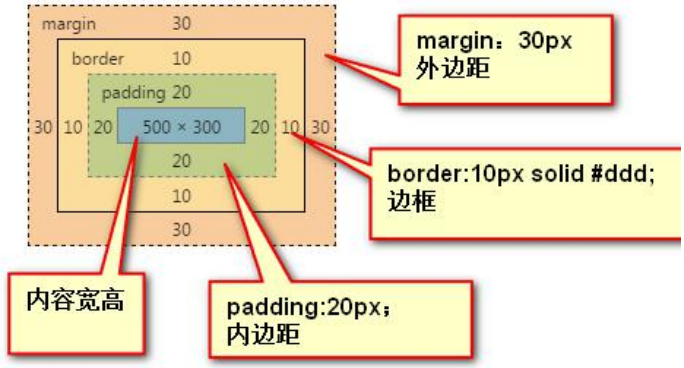
width:内容宽

height：内容高

padding：内边距

border:边框。

```
        .box{
            width:500px;
            height:300px;
            padding:20px;
            border:10px solid #ddd;
            margin:30px;
            background-color: #eee;
        }
```




```
实际宽= 500 + 20*2 + 10*2 = 560px
实际高= 300 + 20*2 + 10*2 = 360px
```

实际工作会告诉我们盒子真正占有的宽高（实际宽高），需要我们计算内容宽和高。

小例子：盒子的宽是 800px，高是 500px,边框是 5px，内边距是 30px，求内容宽和高。

解：

```
内容宽= 盒子宽-边框*2 - 内边距*2
内容高= 盒子高-边框*2 - 内边距*2
```

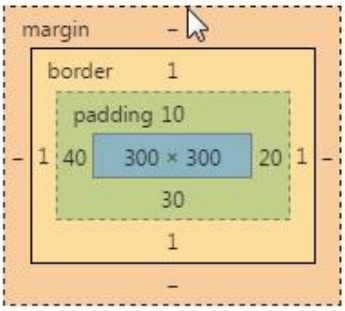
```
width = 800-5*2-30*2=730px
height= 500-5*2-30*2=430px
```

2、padding

padding：内边距。复合属性。

按照方向拆分：

```
            padding-top:10px;
            padding-right:20px;
            padding-bottom:30px;
            padding-left:40px;
```



padding 复合写法：

四值法：上 右 下 左。

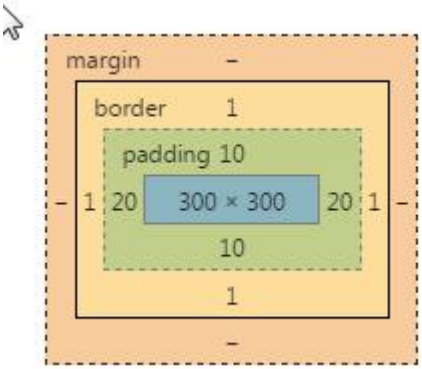
```
padding:10px 20px 30px 40px;
```

三值法：上 左右 下

```
padding:10px 20px 30px;
```

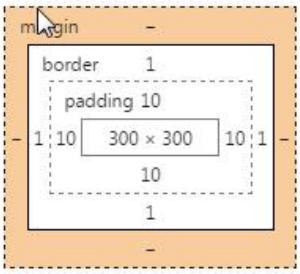
二值法：上下 左右；

```
padding:10px 20px;
```



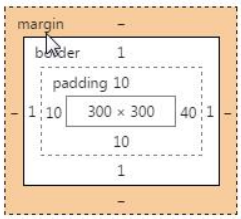
单值法：上下左右

```
padding:10px;
```



实际工作我们设置复合属性，特殊方向的值用单一属性。

```
            padding:10px;
            padding-right:40px;
```



3、margin

margin:外边距。复合属性

四值法：上 右 下 左；

```
margin:10px 20px 30px 40px;
```

三值法：上 左右 下

```
margin:10px 20px 30px;
```

二值法：上下 左右

```
margin:10px 20px;
```

单值法：上下左右

```
margin:10px;
```

按照方向进行划分：

```
margin-top:10px;
margin-right:20px;
margin-bottom:30px;
margin-left:40px;
```

实际工作中先写复合属性，特殊方向的值用单一属性。

3、border

border:边框。复合属性。

①按照方向进行划分：需要把三个类型都写全。

```
border-top:3px solid #ddd;
border-right:3px solid #ddd;
border-bottom:3px solid #ddd;
border-left:3px solid #ddd;
```

②按照类型划分

```
border-width:3px;        //线的宽度
border-style:solid;     //线的类型
border-color:#ddd;      //线的颜色
```

③把以上两种方法结合

```
            border-top-width:10px;
            border-top-color:#ddd;
            border-top-style:dashed;
```

复合写法：

```
border:1px solid #ddd;
```

小例子：

```
        .box{
            width:0;
            height:0;
            border:30px solid #fff;
            border-bottom-color:#f00;
            border-top:none;
        }
```

border-collapse：塌陷

默认值：separate，线毁分开。

collapse,线塌陷，用于制作单线表格。

```
table,tr,td{
    border:1px solid #ddd;
    border-collapse:collapse;
}
```

第1行第1个单元格	第1行第2个单元格	第1行第3个单元格
第2行第1个单元格	第2行第2个单元格	第2行第3个单元格
第2行第1个单元格	第2行第2个单元格	第2行第3个单元格